EP0342838

Publication Title:

User interface for a data input

Abstract:

The user interface system and method disclosed provides a user interface tool for simultaneously selecting a menu item and a value, from a range of values, for the menu item and can be referred to as a valuator menu, since it allows both the selection of a value from a range of values, and the selection of a menu item from a menu list. As a user moves a cursor over a menu of selectable items on a screen display, the item underneath the cursor is highlighted. In addition, as the user moves the cursor within the highlighted menu item, a value relative to the position of the cursor within that menu item is displayed. This valuator value is dynamically updated as the curs
302
or position changes within the menu item. When the user performs an input selection, i.e. through a mouse button or a keyboard interaction, both the selected menu item and the value, relative to the cursor position, are simultaneously returned to the application program running on the data processing system.

.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 342 838 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 89304619.3

(51) Int. Cl.4: G06F 3/033

(22) Date of filing: 08.05.89

The title of the invention has been amended (Guidelines for Examination in the EPO, A-III, 7.3).

(30) Priority: 20.05.88 US 196922

(43) Date of publication of application: 23.11.89 Bulletin 89/47

(84) Designated Contracting States: BE CH DE ES FR GB IT LI NL SE

(71) Applicant: International Business Machines Corporation
Old Orchard Road
Armonk, N.Y. 10504(US)

(72) Inventor: Gest, Stephen Brent
7206 Eganhill Drive
Austin Texas 78745(US)
Inventor: Wymore, Farrell Wade
12148 Jollyville Road, No. 417
Austin Texas 78759(US)

(74) Representative: Blakemore, Frederick Norman et al
IBM United Kingdom Limited Intellectual Property Department Hursley Park
Winchester Hampshire SO21 2JN(GB)

(54) User interface for a data input.

(57) The user interface system and method disclosed provides a user interface tool for simultaneously selecting a menu item and a value, from a range of values, for the menu item and can be referred to as a valuator menu, since it allows both the selection of a value from a range of values, and the selection of a menu item from a menu list. As a user moves a cursor over a menu of selectable items on a screen display, the item underneath the cursor is highlighted. In addition, as the user moves the cursor within the highlighted menu item, a value relative to the position of the cursor within that menu item is displayed. This valuator value is dynamically updated as the cursor position changes within the menu item. When the user performs an input selection, i.e. through a mouse button or a keyboard interaction, both the selected menu item and the value, relative to the cursor position, are simultaneously retumed to the application program running on the data processing system.

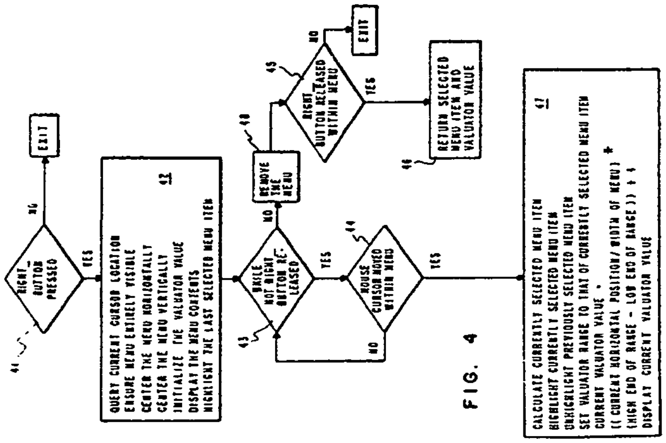

41
RIGHT BUTTON PRESSED
NO
EXIT
YES
42
QUERY CURRENT CURSOR LOCATION
ENSURE MENU ENTIRELY VISIBLE
CENTER THE MENU HORIZONTALLY
CENTER THE MENU VERTICALLY
INITIALIZE THE VALUATOR VALUE
DISPLAY THE MENU CONTENTS
HIGHLIGHT THE LAST SELECTED MENU ITEM
43
WHILE NOT RIGHT BUTTON RE-LEASED
NO
48
REMOVE THE MENU
45
RIGHT BUTTON RELEASED WITHIN MENU
NO
EXIT
YES
46
RETURN SELECTED MENU ITEM AND VALUATOR VALUE
YES
44
MOUSE CURSOR MOVED WITHIN MENU
NO
YES
47
CALCULATE CURRENTLY SELECTED MENU ITEM
HIGHLIGHT CURRENTLY SELECTED MENU ITEM
UNHIGHLIGHT PREVIOUSLY SELECTED MENU ITEM
SET VALUATOR RANGE TO THAT OF CURRENTLY SELECTED MENU ITEM
CURRENT VALUATOR VALUE =
((CURRENT HORIZONTAL POSITION / WIDTH OF MENU) *
(HIGH END OF RANGE - LOW END OF RANGE)) + 4
DISPLAY CURRENT VALUATOR VALUE

FIG. 4

IMPROVEMENTS IN OR RELATING TO USER INTERFACES

The present invention relates to the user interfaces of data processing systems and seeks to provide an improvement in the efficiency thereof as perceived by the user.

A user interface is a sub-system of a data processing system which, at times, is logically owned by the user and, at times, is logically owned by the rest of the system. It includes some means for indicating data to the user - a display - and some user input means by which the user can enter data into the interface - a keyboard, mouse etc. - and data and control interconnection with the rest of the system. Data entered by the user is reflected in the display and becomes available to the rest of the system when the user relinquishes ownership of the interface by some particular keystroke or mouse manipulation. The classical dialog between the user and the system can be thought of as a sequence of questions asked by the system and answers supplied thereto by the user, the questions and answers being sequentially dependent, though, frequently, so-called "fastpaths" have been provided by which an expert user can answer a sub-sequence of questions, the asking of which is implied, by a single composite data entry which the user has to foresee from his experience and construct. The present invention seeks to provide increased interface efficiency by providing the less expert user with a built-in fastpath facility in that it provides logic on the system side of the interface for presenting to the user a display indicating a plurality of areas and for interpreting the selection of a position, by the user, in one of the areas displayed as a pair of data entries, one derived from the area selected and the other derived from both the area selected and the position selected in that area. In this way, what is perceived by the user as a single data entry can represent, for example, probability automatically as a numerical value between 0 and 1, line length automatically as a number of characters between 32 and 132, an identified menu by name at some dependent level below the entry level, or whatever. These examples are given only to indicate the kind of flexibility provided by this arrangement. Other examples are given hereinafter. Preferably, the logic will augment the display to indicate the current potential selection to the user in plain terms.

Looking at this matter another way, in more detail, windows, icons, mouse interactions, and pop-up menu systems are part of a computer user interface known in the art. A typical user interface may include a valuator tool. A valuator tool is a representation of a gauge which graphically presents a value over some range of values. As shown in Fig. 1, a dial 11 and a slide bar 12 are examples of valuator tools. The user of such a tool can graphically manipulate the valuator to indicate different values. The valuator tool returns, upon manipulation, a value with respect to its valuator type. For example, a volume control device could be represented by a slide bar 12 with graduated increments along the bar, as shown. Manipulating the slide bar returns the specified increment along the bar, thus increasing or decreasing the volume of the device.

A menu tool 14, as shown in Fig. 1, is a window containing a series of selectable items that appears when a specific mouse button(s) is pressed. The menu may appear at the current location of the mouse cursor, ("pop-up" menu), or appear below the menu's title when the user presses a mouse button(s) within that title, ("pull-down" menu).

Menu items are words or phrases that describe some type of operation that the application can perform. The user selects an item from the menu by invoking the menu, (pressing and holding down a mouse button(s)), moving the mouse cursor so that it points to the item to be selected, and releasing the mouse button(s). As the mouse cursor moves from menu item to menu item, the item currently being pointed to by the cursor is highlighted in some fashion, usually reverse video. Releasing the mouse button(s) on a highlighted item selects that item and causes the application to perform that operation.

The problem with current technology is that often it is necessary to provide parameters for operations selected from a menu. Currently, these parameters are provided by invoking some dialog between the user and the application either before or after the selection is made. Dialogs are invoked before selections are made when all the operations for a given menu possess some common subset of parameters. Users supply the requested parameters via a dialog of additional keystrokes and/or mouse actions. For a commonly performed operation, this dialog is both annoying and time consuming for the user of the application.

According to the present invention there is provided a user interface for or as a sub-system of a data processing system includes display means for indicating data to the user, user input means by which the user can enter data into the interface, data and control interconnection means to the processing system, logic on the system side of the interface for presenting to the user a display indicating a plurality of areas and for interpreting the selection of a position, by the user, in one of the areas displayed, as a pair of data entries, one derived from the area selected and the other derived from both the area selected and the position selected in that area.

The present invention equally provides a method of communicating with a data processing system via a user interface thereof which includes display means for indicating data to the user, user input means by which the user can enter data into the interface, data and control interconnection means to the processing system, comprising

presenting to the user a display indicating a plurality of areas and

interpreting the selection of a position, by the user, in one of the areas displayed, as a pair of data entries, one derived from the area selected and the other derived from both the area selected and the position selected in that area.

In this way it is possible to economise the user's interaction in specifying multiple pieces of information to an application running on a processing system and to reduce the number of input selections, such as through the number of keystrokes on a keyboard or through mouse movement and button selection, required by a user in a user interface.

In an embodiment of the invention disclosed hereinafter, the valuator menu addresses the above mentioned problems by providing a convenient way for the user to specify multiple pieces of information to the application with a reduced number of interactions.

The valuator menu combines the valuator tool and the menu tool to create a unique interactive tool from which the user can specify multiple pieces of information to the application program, running on the processing system, in an effective and economical manner. The user specifies this information to an application by selecting a choice from a menu, while simultaneously selecting a value, from a range of values, specific to each menu choice.

The user selects a menu item by moving the mouse cursor vertically over the menu in the manner described above. The user can additionally select a valuator value, from a range of values, by moving the cursor horizontally over the menu. The current valuator value is displayed within the menu, and is dynamically updated with respect to the mouse cursor's horizontal position within the menu item. The valuator value may be in any range, and adjusted to any appropriate scale.

Releasing the mouse button(s) over a highlighted menu item returns multiple pieces of information to the application via a single mouse operation. The information returned consists of the selected menu operation to be performed, and a valuator value pertaining to that operation.

The present invention will be described further by way of example with reference to embodiments thereof as illustrated in the accompanying drawings, in which:-

Fig. 1 illustrates a dial valuator tool, a slide bar valuator tool, and a pop-up menu, all known in the art as graphical user interface tools;

Fig. 2A illustrates the hardware, including a display, of a processing system;

Fig. 2B illustrates the logical structure of the processing system;

Fig. 2C illustrates the physical structure of the processing system;

Fig. 2D illustrates the software components thereof;

Fig. 3 illustrates a screen display output of the valuator menu;

Fig. 4 illustrates a flow chart of the logic for implementing the valuator menu;

Fig. 5A - 5F illustrates various screen displays showing the function of the valuator menu in simultaneously selecting one form of menu item and a value for that menu item;

Fig. 6A - 6D illustrates various screen displays showing the function of the valuator menu in simultaneously selecting another form of menu item and a value therefor; and

Fig. 7A - 7C illustrates various screen displays showing the function of the valuator menu in simultaneously selecting a further menu item and a value therefor.

The disclosed enhancement was targeted for the IBM RT PC 2, running the AIX operating system 5. For more information on the RT PC, and the AIX operating system, reference can be made to the following. Bach, M.J., The Design of the UNIX Operating System, Prentice Hall, 1986. Lang, T.G. and Mothersole, T.L., Design of the RTPC VRM Nucleus, September 1, 1986. AIX Operating System Commands Reference, IBM Corporation, SC23-0790. AIX Operating System Managing the AIX Operating System, IBM Corporation, SC23-0793. AIX Operating System Programming Tools and Interfaces, IBM Corporation, SC23-0789. AIX Operating System Technical Reference, Volumes 1 and 2, IBM Corporation, SC23-0808 and SC23-0809. IBM RT Personal Computer Technology, IBM Corporation, SA23-1057, 1986. Virtual Resource Manager Technical Reference, Volumes 1 and 2, IBM Corporation, SC23-0816 and SC23-0817.

The valuator menu user interface tool detailed hereinafter is a vehicle with which a user can specify several pieces of information for a commonly used operation with a minimal amount interaction. It comes in various forms which depend, for their detail on the nature of the system to which they interface. The initial form was designed to enhance the "Xdbx" interface to the "dbx" symbolic debugger 9 to improve the general presentation and usability thereof. The dbx and plain Xdbx symbolic debuggers are described in the IBM RT PC Programming Tools and Interfaces, Version 2.2, IBM Corporation. The features of this invention were, at the time of filing the parent application, intended to become a part of the IBM RT PC Advanced Interactive Operating System Extended Services Program.

As shown in Fig. 2D the Xdbx debugger interface 13 and the dbx symbolic debugger 9, along with X-Windows 10, are application development products 8 as shown in Fig. 2C.

The interface 13 makes use of the X-Windows system 10, which supplies most of the primitive window management tools that allow more complex tools, like the valuator menu, to be built. More information on X-Windows can be found in IBM RT PC X-Windows Version 1.1, X-Windows User Guide & Reference, Sept. 1987. The Xdbx debugger 13 allows the user to specify each dbx operation 9, giving the user complete functionality, without being constrained by each operation's syntax.

One commonly used operation in the dbx symbolic debugger 9 is the ability to list assembly instructions 22, Fig. 3, or display the contents of an address 26 as shown by contents 24. These operations require three pieces of information: an address 26, the number of memory items to display 20, and a mode specifying how to display the memory 110. As shown in Fig. 3, Dbx supports the following display modes: string 101, octal byte 102, ascii byte 103, short decimal 104, long decimal 105, single precision float 106, double precision float 107, short octal 108, long octal 109, short hexadecimal 110, long hexadecimal 111, and machine instruction 112.

With reference to both Fig. 3 and Fig. 4, the valuator menu tool allows the user to specify the above information to Xdbx in an economical and effective manner. The user selects the starting address 26 of the range of memory to display by moving the mouse cursor 18 so that it points to that address 26 and clicking the left mouse button 16, Fig. 2A. The selected address 26 would then be highlighted in reverse video as shown.

Having selected the starting address 26, the user could invoke the valuator menu by pressing and holding down the right mouse button, step 41 Fig. 4. As shown in Fig. 3, the user moved the cursor to another location on the screen to enhance clarity of Fig. 3 before invoking the valuator menu. However, this is not necessary, as the valuator menu 31 could appear at location 26.

Nevertheless, the valuator menu 31 appears centred horizontally about the current mouse cursor location, step 42. The valuator menu 31 is positioned vertically such that the cursor appears over the same menu item 110 that was previously selected, or over the first menu item 101, if it is the first time that the valuator menu 31 is invoked.

The user can simultaneously select a menu item, i.e. a display mode 101-112, and one of a number of values, i.e. number of memory items 20, to be displayed. The user could select a display mode 101-112 by moving the mouse cursor vertically over the valuator menu 31. As the mouse cursor moves from menu item to menu item, the item 110 currently being pointed to by the cursor 18 is highlighted in reverse video, step 42, Fig. 4.

Simultaneously, the user could select one of a number of values, i.e. memory items 20, to be displayed by moving the cursor horizontally over the valuator menu 31, step 44. As the user moves the cursor horizontally within the menu 31, the valuator value 20 is updated with respect to the cursor's horizontal position, step 47. Since the menu is initially centred about the mouse cursor, the initial value represents a mid range value of the range specified.

The menu items 101 - 112 would consist of the possible memory display modes, while the valuator value 20 would represent the current number of memory items to display. Releasing the right button over a highlighted display mode, step 45, returns both that display mode 110, and the number of memory items to display 20, back to the Xdbx program via one single mouse button release, step 46. With this information, Xdbx could now invoke the dbx operation to display a memory range 24, preserving the dbx syntax as shown by output 28, Fig. 3. The dbx syntax "0x100001c0/17x" means that beginning at memory location 0x100001c, seventeen 16 bit integers, i.e. short integers, will be displayed as hexadecimal numerals. These seventeen integers are shown as numeral 24 in Fig. 3.

For this commonly used operation, the valuator menu was the perfect vehicle to allow the user to specify multiple pieces of information in an efficient and economical fashion.

The following flow-chart, in program design language, illustrates the logic of the above operation:

```
if ( right_button pressed ) {
    query the current mouse cursor location;
    adjust the menu such that it is entirely visible;
    centre the menu horizontally about the current
          mouse cursor location;
    centre the menu vertically about the last selected
          memory display mode;
    set the valuator_value to .5 * valuator_range;
    display the menu title, menu items, and
     valuator_value;
    highlight the last selected memory display mode in
          reverse video;

    while ( not right_button released ) {
        if ( mouse cursor moved within menu ) {
            calculate currently selected memory
               display mode;
            highlight currently selected memory
               display mode;
            unhighlight previously highlighted memory
               display mode;
            calculate the current valuator_value =
                   (( current horizontal position /
                   width of menu ) * (upper
                   range-lower range))
                   + 1;
           display current valuator_value represent-
                   ing number of memory items to
                   display;
       }
   }

   /* right_button released */

   remove the menu;
   if ( right_button released within menu )
         display the number of memory items indicated
              by the valuator_value in the selected
              memory mode;
}
```

The program design language code listed above is specific to this preferred embodiment. However, the valuator menu of the present invention is not limited to the specific application as described in reference to the previous embodiment employing the Xdbx debugger. For a more general embodiment, references to memory items, display modes, and memory modes would be replaced by other words representing a different embodiment. Generally, for other embodiments, the selected action will be implemented with the valuator value as its argument.

The valuator menu technique can be implemented in a host of other applications. For example, this tool can be used to specify a certain hue of a particular colour. Fig. 5A through Fig. 5F illustrate sequences of a menu as a cursor is moved to simultaneously specify a certain hue of a particular colour. The valuator menu items 51-58 could consist of the range of possible colours starting with white 51 and ending with black 58. The valuator value 20 would represent the amount of saturation for the colour currently being pointed to by the mouse cursor, which is highlighted in reverse video. For this example, the valuator value 20 might range from 0 to 1 in increments of hundredths, indicating the saturation of the colour. Moving vertically over the menu would select a colour, while moving horizontally within the menu would change the amount of saturation for that particular colour.

As shown in Fig. 5A, the cursor 18 is pointing to colour white 51 at the farthest left position indicating a zero saturation level as shown by the valuator value 20. In Fig. 5B the cursor is moved down vertically to the yellow colour item 52. As the cursor 18 is also moved horizontally to the right, the valuator value 20 changes dynamically with the horizontal position indicating a change in the value of the saturation level that may be selected by the user. Fig. 5C illustrates the dynamic change in the valuator value 20 as the cursor 18 location changes its relative horizontal position. Fig. 5D, 5E, and 5F also illustrate this dynamic change in the valuator value 20 as the cursor 18 location changes its relative horizontal position.

In this manner, the user could select simultaneously a colour item 51-58 and one of a plurality of saturation levels for the colour item, in one click of the mouse button. The following flow-chart, in program design language code, further illustrates the logic of simultaneous selection of two items with one user input action.

```
if ( right_button pressed ) {
  query the current mouse cursor location;
  adjust the menu such that it is entirely visible;
  centre the menu horizontally about the current
     mouse cursor location;


 centre the menu vertically about the last selected
    colour;
 set the saturation value to .5 * saturation range;
 display the menu title, menu colour items, and
    saturation value;
 highlight the last selected colour item in reverse
    video;

 while ( not right_button released ) {
   if ( mouse cursor moved within menu ) {
     calculate currently selected colour item;
     highlight currently selected colour item;
     unhighlight previously highlighted colour
       item;
     calculate the current saturation value =
         (( current horizontal position /
         width of menu ) * saturation
         range ) + 1;
     display current saturation value;
   }
 }

 /* right_button released */

 remove the menu;
 if ( right_button released within menu )
    return selected colour and saturation value
    to the application;
}
```

Another embodiment of the valuator menu is as a user interface tool that allows the user to simultaneously select a font style and a character pitch for that font. Fig 6A through Fig. 6D illustrates a possible sequence of a visual representation on a display as the cursor 18 is moved throughout the menu 31. Figures 6A -6D also illustrate that the menu items 61-66 can be listed horizontally instead of vertically as previously shown. Also, the selection of the pitch, i.e. the valuator value, could also occur relative to the same horizontal direction as the menu items 61-66. As the cursor 18 is moved into a new menu item, the valuator value 20 would reset for the new menu item.

In this embodiment, as shown in Figs. 6A - 6D, the valuator menu items 61-66 would represent the possible font styles, while the valuator value 20 would represent the possible character pitches for each font style. This range of character pitches could be different for each font. Moving over the menu 31 would select a different font style, while simultaneously changing the range of the valuator values 20, i.e. character pitches, associated with the selected font style. Moving the cursor within the menu item would update the value with the possible character pitches for that particular font style.

As shown in Fig. 6A, the cursor location indicates menu item 61 in reverse video, with a character pitch of 10, as indicated by the valuator value 20. As the cursor 18 position is moved horizontally, (the vertical positioning of the cursor 18 is not relevant in this embodiment), the character pitch changes to 24 as indicated by valuator value 20 in Fig. 6B. As shown in Fig. 6C, the Roman font style 65 is indicated in reverse video by the cursor 18 position, along with the 24 character pitch as indicated by valuator value 20. As the cursor 18 position is moved horizontally to the right as shown in Fig. 6C, the valuator value resets relative to the next menu item 66. The relative cursor 18 position within menu item 66, Fig. 6D, indicates simultaneously the menu item 66 is to be selected along with a character pitch of 12.

The following flow-chart,in program design language code, illustrates the logic of this embodiment as shown in Fig. 6A - 6D.

```
if ( right_button pressed ) {
    query the current mouse cursor location;
    adjust the menu such that it is entirely visible;
    centre the menu horizontally about the current
          mouse cursor location;
    centre the menu vertically about the last selected
          font style item;
    set the character pitch value to .5 * pitch range
          of last selected font style item;
    display the menu title, font style items, and
          character pitch value;
    highlight the last selected font style item in
          reverse video;

    while ( not right_button released ) {
        if ( mouse cursor moved within menu ) {
            calculate currently selected font style
               item;
            highlight currently selected font style
               item;
            unhighlight previously highlighted font
               style item;
            set character pitch range to that of the
               currently selected font style item;
            calculate the current character pitch
               value =
                   (( current horizontal position /
                    width of menu ) * pitch range ) +
                    1;
            map character pitch value to pitch value
               range
            display current mapped character pitch
               value;
        }
        }

        /* right_button released */

        remove the menu;
        if ( right_button released within menu )
             return selected font style and character
             pitch value to the application;
    }
```

A further embodiment of the present invention employs a calendar menu as shown in Fig 7A - 7C. With the user interface of the present invention, a user could check the plans for a particular day of the year in an economical manner. The valuator menu items 71-82 would represent the months of the year, while the valuator value 20 would represent the days for each month. The range of days per month would obviously change per month, as shown in Fig. 7A and Fig. 7B with respect to the same relative position of the cursor 18 in the menu item, and the different resulting valuator value 20.

As shown in Fig. 7A - 7C, moving the cursor 18 vertically over the menu 31 would select a menu item 71 -82 in reverse video. The range of the valuator value 20 would change relative to the highlighted menu item. Moving the cursor 18 horizontally within the menu 31 would update the value 20 displayed representing the day for the particular month highlighted. With this information, an application could display the user's agenda for that particular day of the year. The user simultaneously selected a menu item (month), and a value (a particular day) of that selected menu item with one user input selection.

The following flow-chart, in program design language code, illustrates the logic of this last embodiment.

```
if ( right_button pressed ) {
    query the current mouse cursor location;
    adjust the menu such that it is entirely visible;
    centre the menu horizontally about the current
          mouse cursor location;
    centre the menu vertically about the last selected
          month item;
    set the day of the month value to .5 * range of
          days of last selected month item;
    display the menu title, month items, and day
          value;
    highlight the last selected month item in reverse
          video;

    while ( not right_button released ) {
        if ( mouse cursor moved within menu ) {
            calculate currently selected month item;
            highlight currently selected month item;
            unhighlight previously highlighted month
               item;
            set range of days to that of the currently
               selected month item;
            calculate the current day value =
                   (( current horizontal position /
                    width of menu ) * range of days )
                    + 1;
            display current day value;
        }
    }

    /* right_button released */

    remove the menu;
    if ( right_button released within menu )
```

```
            return selected month and day value to the
            application;
    }
```

While the invention has been particularly shown and described with reference to a preferred embodiment and other embodiments, it will be understood by those skilled in the art that various changes in form and detail may be made without departing from the scope of the appended claims. For example, various changes may include, but are not limited to, the interchangeability of references to either a horizontal direction or a vertical direction. In addition, a keyboard interaction or other input device can be used instead of using the mouse input device as described herein. Also, the additional embodiments were shown as examples of the various uses of the present invention. It would be within the scope of this invention to implement the valuator menu of this invention in other embodiments not specifically discussed herein.

Claims

1. A user interface for or as a sub-system of a data processing system includes display means for indicating data to the user, user input means by which the user can enter data into the interface, data and control interconnection means to the processing system, logic on the system side of the interface for presenting to the user a display indicating a plurality of areas and for interpreting the selection of a position, by the user, in one of the areas displayed, as a pair of data entries, one derived from the area selected and the other derived from both the area selected and the position selected in that area.

2. A user interface as claimed in claim 1, wherein interpretation of the areas presented collectively comprises a displayable list of user selectable actions and the pair of data items simultaneously selectable is one of the selectable actions and a value within a range of values of that selectable action.

3. A user interface as claimed in claim 2, including cursor logic responsive to the user input means by which, moving a cursor in a first direction selects one of the plurality of selectable actions and moving the cursor in a second direction within the selected item selects the value of the action.

4. A user interface as claimed in claim 3, including indicator logic, responsive to moving the cursor in the second direction to cause the display of a dynamically updated indicator representing the value associated with the current location of the cursor in the second direction.

5. A user interface as claimed in claim 4, further comprising means for simultaneously specifying, to an application running on the processing system, the selected action and the correlated value of the selected action, relative to the position of the cursor, by a single user input interaction.

6. A user interface as claimed in any preceding claim, wherein the generated display comprises a menu displaying a plurality of selectable colour item choices enabling the simultaneous selection of a colour and a saturation value within a range of a plurality of saturation values for that one of the plurality of selectable colour items to the cursor position within the selectable colour item.

7. A user interface as claimed in any of claims 1 to 5, wherein the generated display comprises a menu displaying a plurality of selectable font style item choices enabling the simultaneous selection of a font style and a character pitch value within a range of a plurality of character pitch values for that one of the plurality of selectable font style items.

8. A method of communicating with a data processing system via a user interface thereof which includes display means for indicating data to the user, user input means by which the user can enter data into the interface, data and control interconnection means to the processing system, comprising
presenting to the user a display indicating a plurality of areas and
interpreting the selection of a position, by the user, in one of the areas displayed, as a pair of data entries, one derived from the area selected and the other derived from both the area selected and the position selected in that area.

9. A method as claimed in claim 8, wherein the display is a menu having a plurality of user selectable actions, the first data item being one of the selectable actions of the menu and the second data item being a value within a range of a plurality of values for that selectable action according to the cursor position within the selectable action, simultaneous selection being made by moving a cursor in a first direction to select the action and in a second direction to select the value therefor.

10. A method as claimed in claim 9, further including displaying a dynamically updated indicator representing the value of the selected action associated with the current location of the cursor.

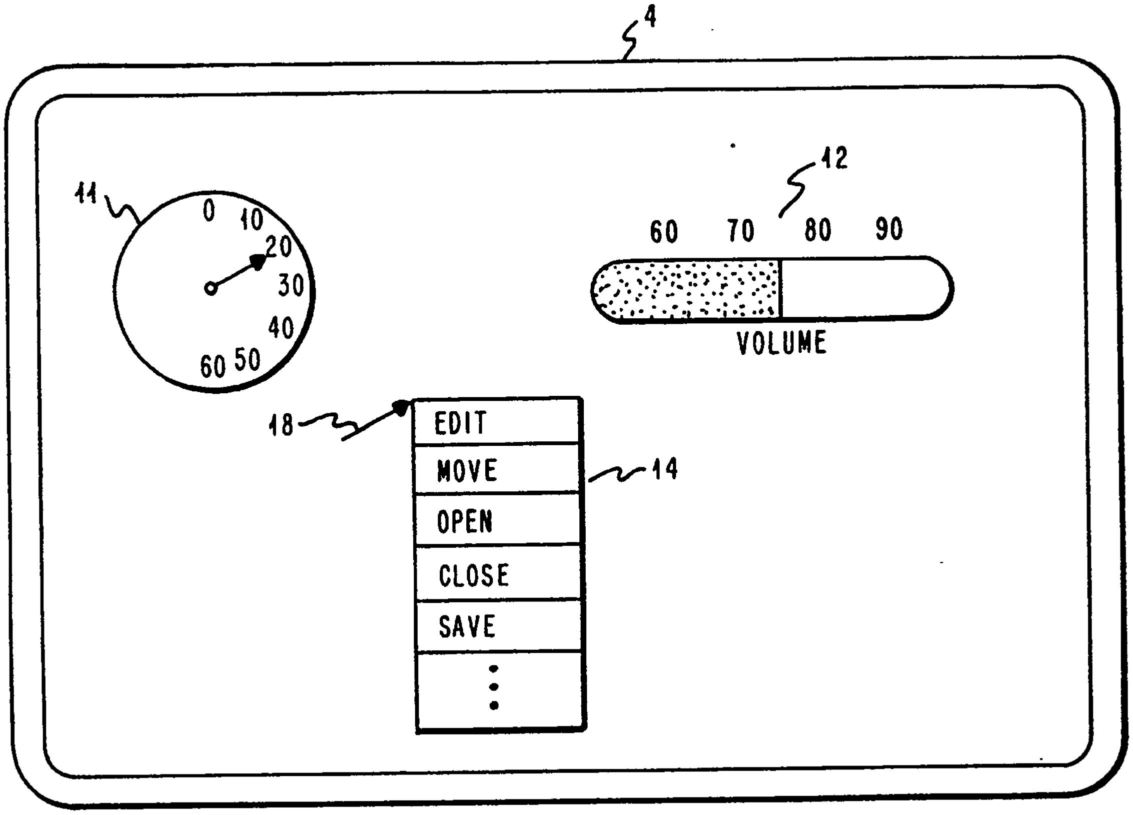

4
11
0
10
20
30
40
50
60
12
60
70
80
90
VOLUME
18
EDIT
MOVE
OPEN
CLOSE
SAVE
14

FIG. 1

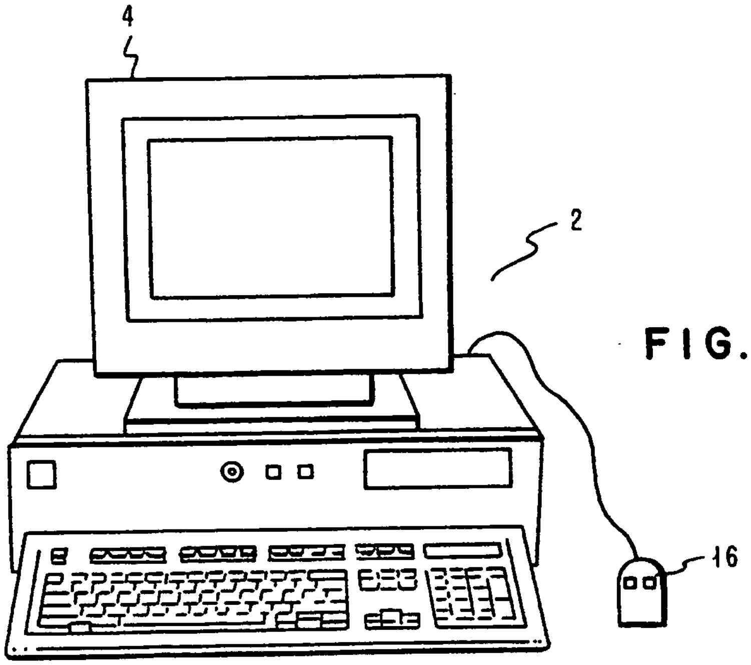

4
2
16

FIG. 2A

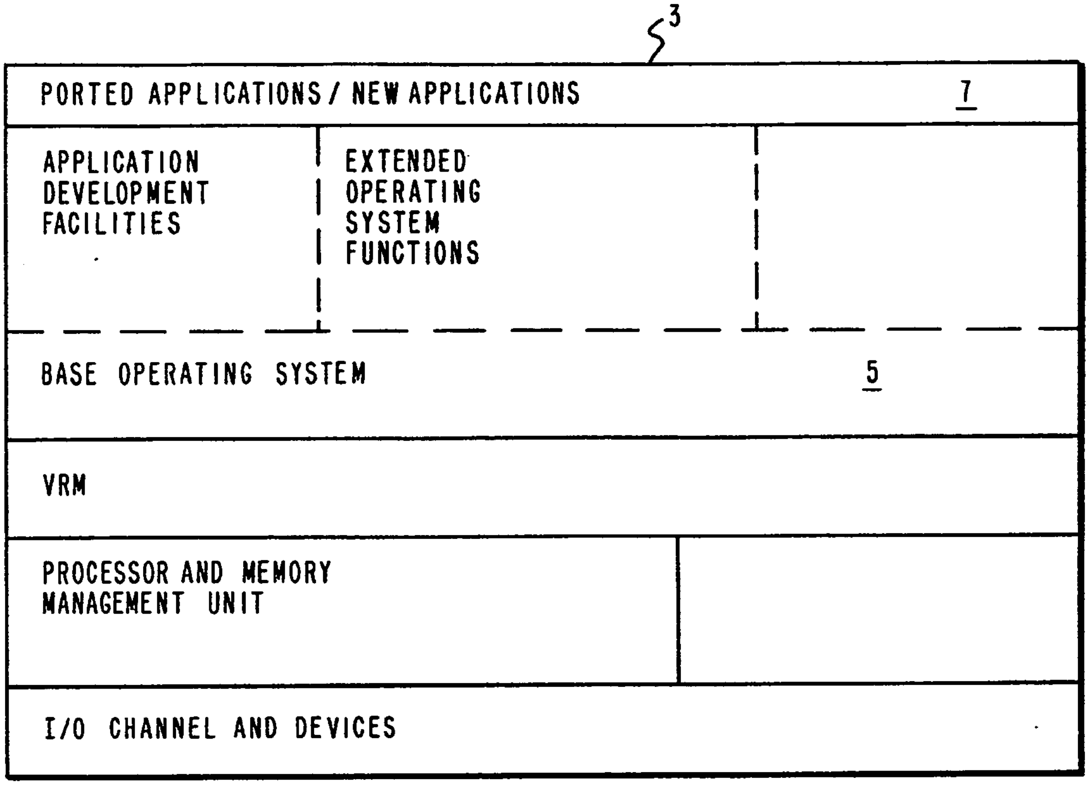

3
PORTED APPLICATIONS / NEW APPLICATIONS
7
APPLICATION DEVELOPMENT FACILITIES
EXTENDED OPERATING SYSTEM FUNCTIONS
BASE OPERATING SYSTEM
5
VRM
PROCESSOR AND MEMORY MANAGEMENT UNIT
I/O CHANNEL AND DEVICES

FIG. 2B

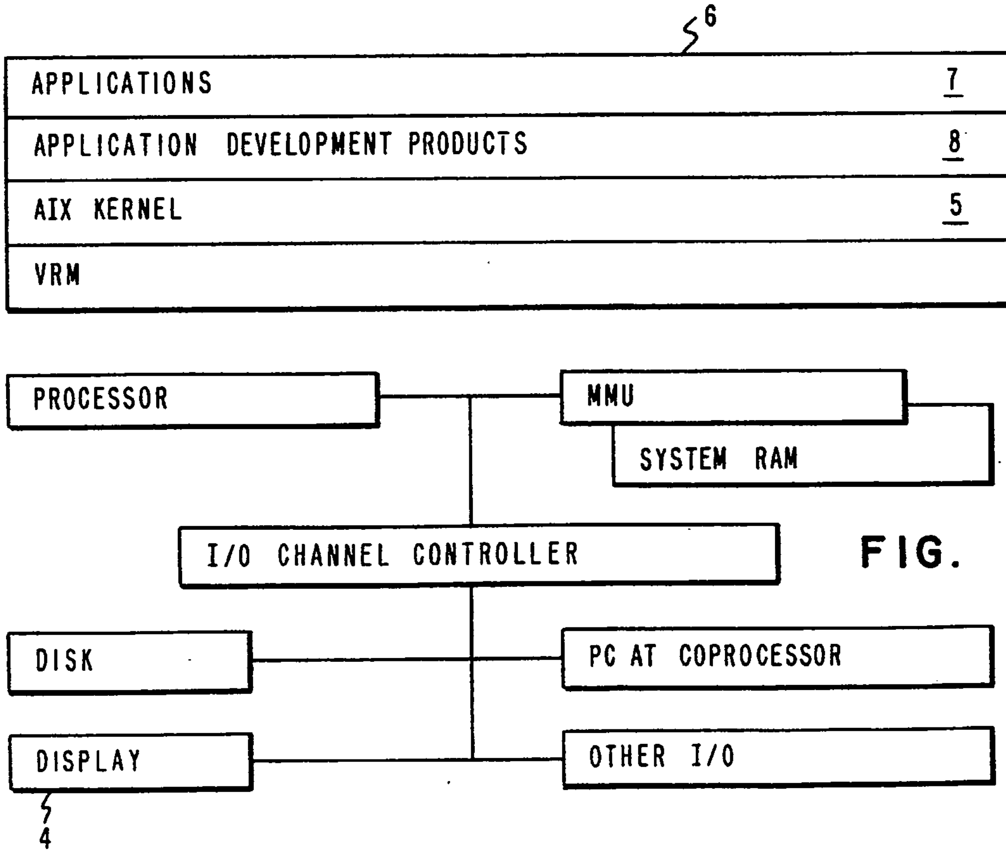

6
APPLICATIONS
7
APPLICATION DEVELOPMENT PRODUCTS
8
AIX KERNEL
5
VRM
PROCESSOR
MMU
SYSTEM RAM
I/O CHANNEL CONTROLLER
DISK
PC AT COPROCESSOR
DISPLAY
OTHER I/O
4

FIG. 2C

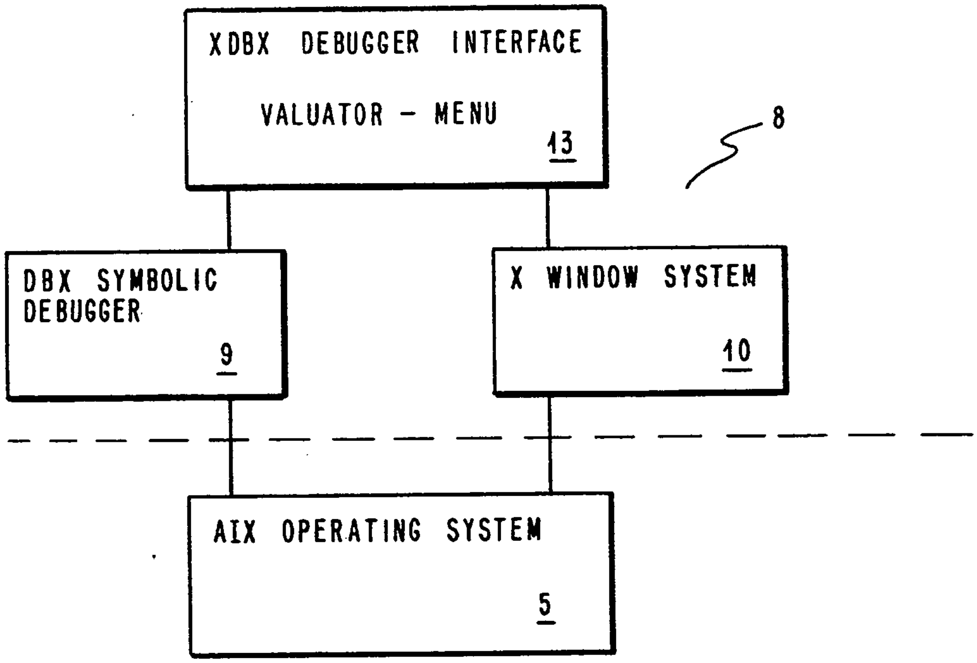
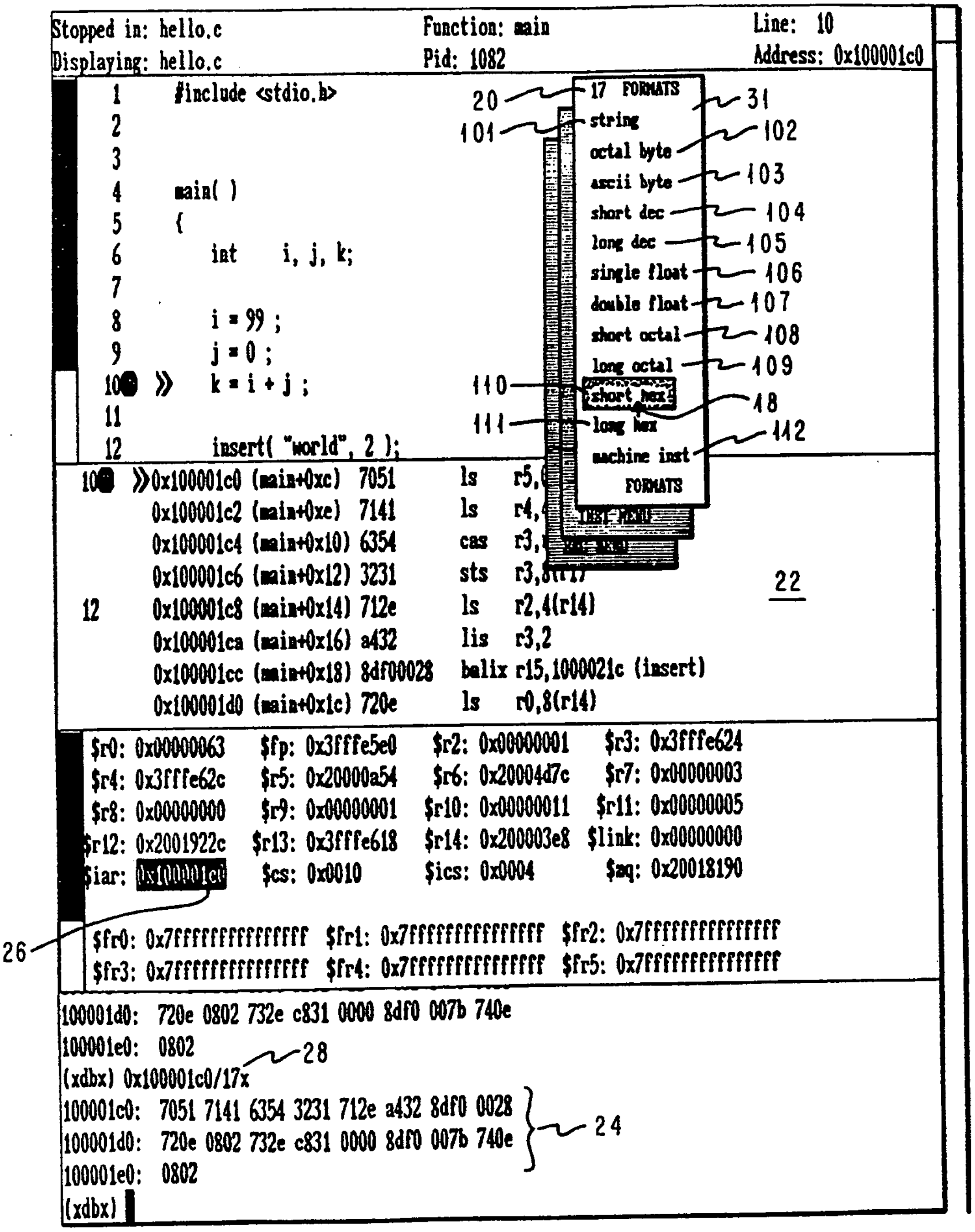

XDBX DEBUGGER INTERFACE
VALUATOR - MENU
13
8
DBX SYMBOLIC DEBUGGER
9
X WINDOW SYSTEM
10
AIX OPERATING SYSTEM
5

FIG. 2D



```
Stopped in: hello.c                    Function: main          Line:  10
Displaying: hello.c                    Pid: 1082               Address: 0x100001c0

 1      #include <stdio.h>
 2
 3
 4      main( )
 5      {
 6          int     i, j, k;
 7
 8          i = 99 ;
 9          j = 0 ;
10●  »     k = i + j ;
11
12          insert( "world", 2 );

10●  »0x100001c0 (main+0xc)  7051      ls    r5,
      0x100001c2 (main+0xe)  7141      ls    r4,
      0x100001c4 (main+0x10) 6354      cas   r3,
      0x100001c6 (main+0x12) 3231      sts   r3,8(r1
12    0x100001c8 (main+0x14) 712e      ls    r2,4(r14)
      0x100001ca (main+0x16) a432      lis   r3,2
      0x100001cc (main+0x18) 8df00028  balix r15,1000021c (insert)
      0x100001d0 (main+0x1c) 720e      ls    r0,8(r14)

 $r0: 0x00000063   $fp: 0x3fffe5e0   $r2: 0x00000001   $r3: 0x3fffe624
 $r4: 0x3fffe62c   $r5: 0x20000a54   $r6: 0x20004d7c   $r7: 0x00000003
 $r8: 0x00000000   $r9: 0x00000001  $r10: 0x00000011  $r11: 0x00000005
$r12: 0x2001922c  $r13: 0x3fffe618  $r14: 0x200003e8 $link: 0x00000000
$iar: 0x100001c0   $cs: 0x0010      $ics: 0x0004       $mq: 0x20018190

 $fr0: 0x7fffffffffffffff  $fr1: 0x7fffffffffffffff  $fr2: 0x7fffffffffffffff
 $fr3: 0x7fffffffffffffff  $fr4: 0x7fffffffffffffff  $fr5: 0x7fffffffffffffff

100001d0:  720e 0802 732e c831 0000 8df0 007b 740e
100001e0:  0802
(xdbx) 0x100001c0/17x
100001c0:  7051 7141 6354 3231 712e a432 8df0 0028
100001d0:  720e 0802 732e c831 0000 8df0 007b 740e
100001e0:  0802
(xdbx)
```

FIG. 3

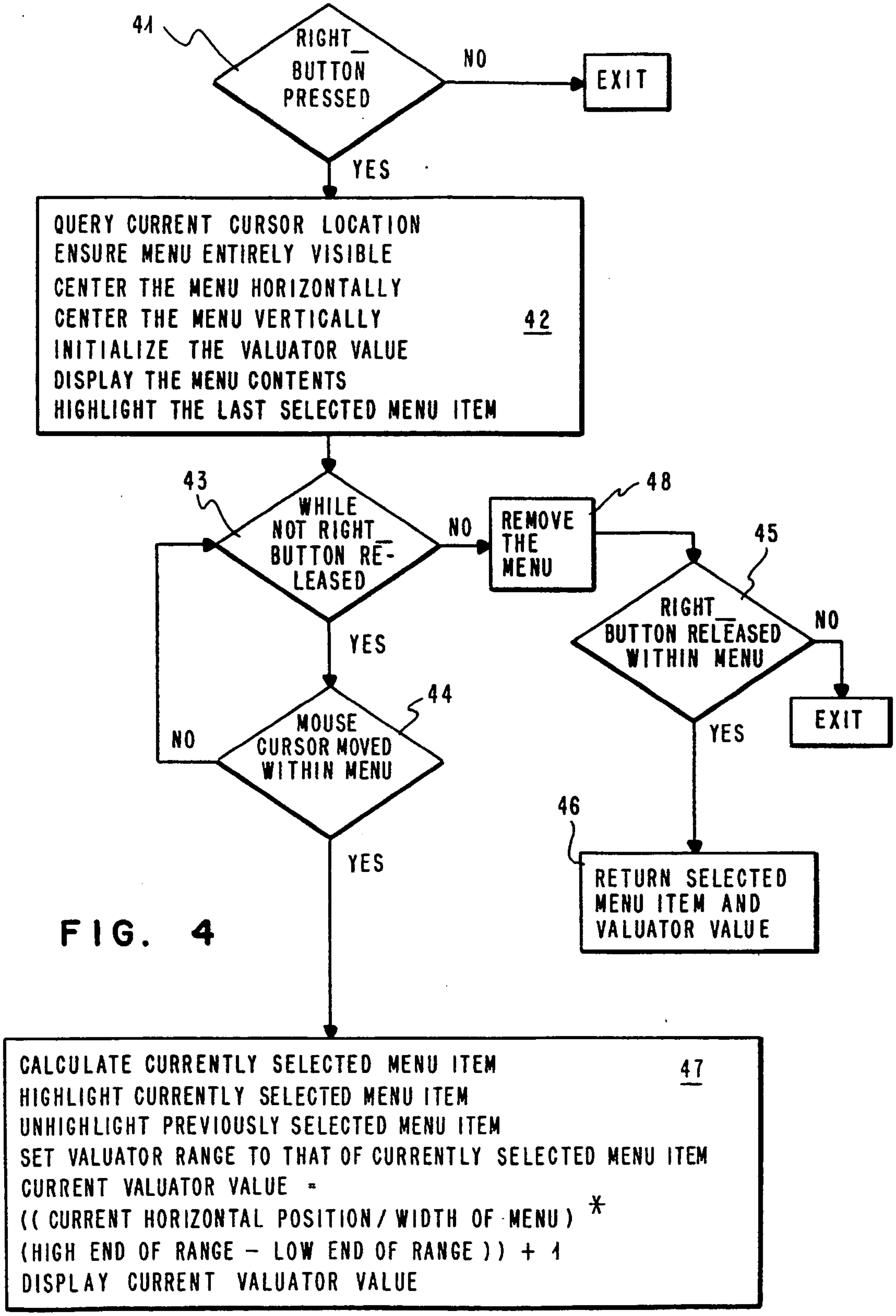

41
RIGHT_ BUTTON PRESSED
NO
EXIT
YES
QUERY CURRENT CURSOR LOCATION
ENSURE MENU ENTIRELY VISIBLE
CENTER THE MENU HORIZONTALLY
CENTER THE MENU VERTICALLY
INITIALIZE THE VALUATOR VALUE
DISPLAY THE MENU CONTENTS
HIGHLIGHT THE LAST SELECTED MENU ITEM
42
43
WHILE NOT RIGHT_ BUTTON RE-LEASED
NO
REMOVE THE MENU
48
45
RIGHT_ BUTTON RELEASED WITHIN MENU
NO
EXIT
YES
YES
44
MOUSE CURSOR MOVED WITHIN MENU
NO
YES
46
RETURN SELECTED MENU ITEM AND VALUATOR VALUE
CALCULATE CURRENTLY SELECTED MENU ITEM
47
HIGHLIGHT CURRENTLY SELECTED MENU ITEM
UNHIGHLIGHT PREVIOUSLY SELECTED MENU ITEM
SET VALUATOR RANGE TO THAT OF CURRENTLY SELECTED MENU ITEM
CURRENT VALUATOR VALUE =
((CURRENT HORIZONTAL POSITION / WIDTH OF MENU) *
(HIGH END OF RANGE - LOW END OF RANGE)) + 1
DISPLAY CURRENT VALUATOR VALUE

FIG. 4

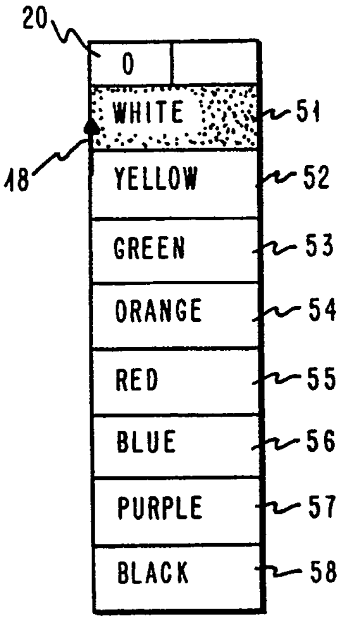


FIG. 5A

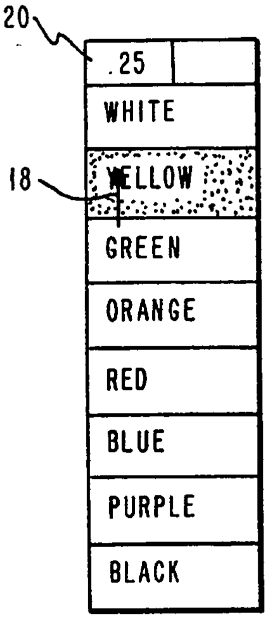


FIG. 5B

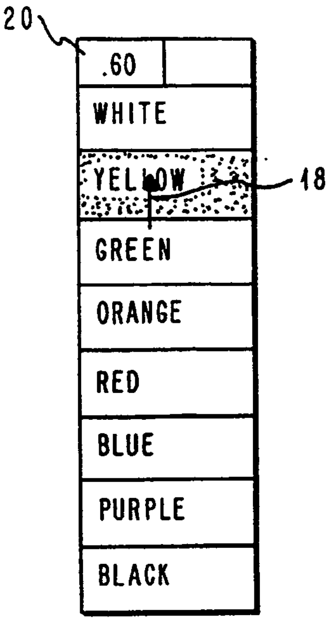


FIG. 5C

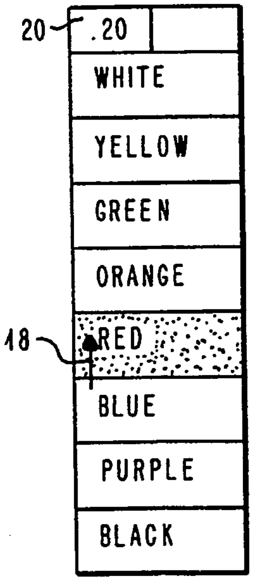


FIG. 5D

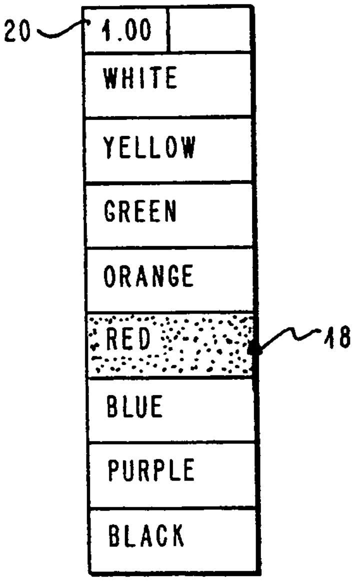


FIG. 5E

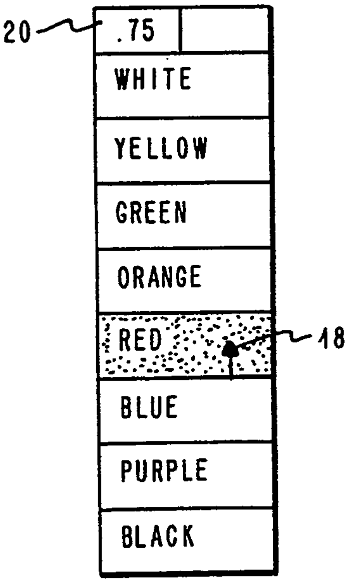


FIG. 5F

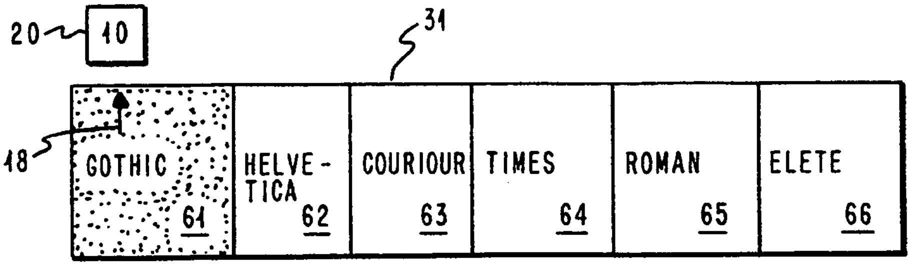


FIG. 6A

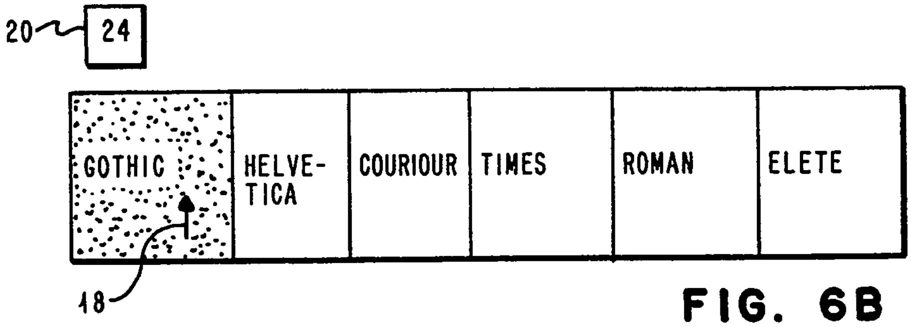


FIG. 6B

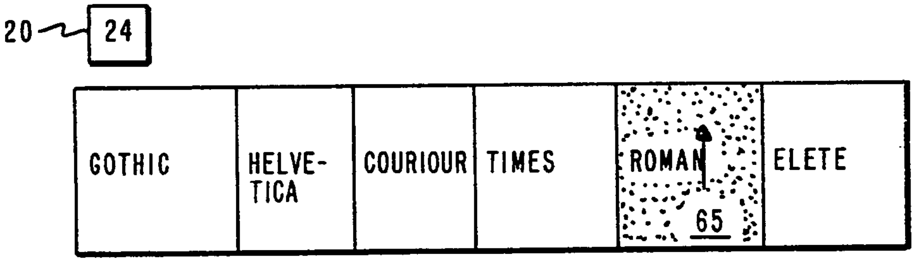


FIG. 6C

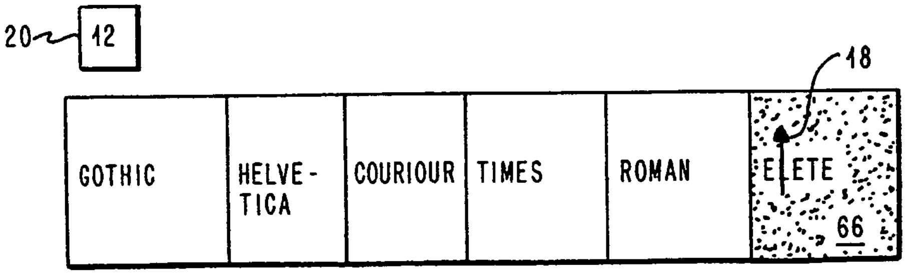


FIG. 6D

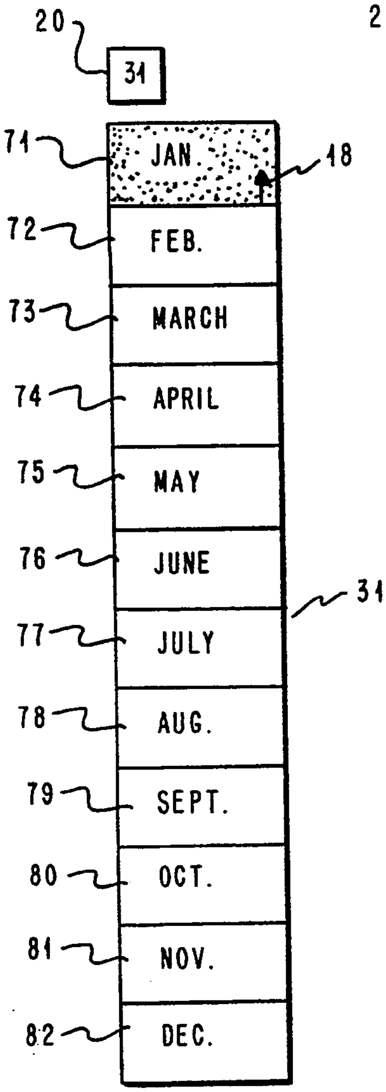


FIG. 7A

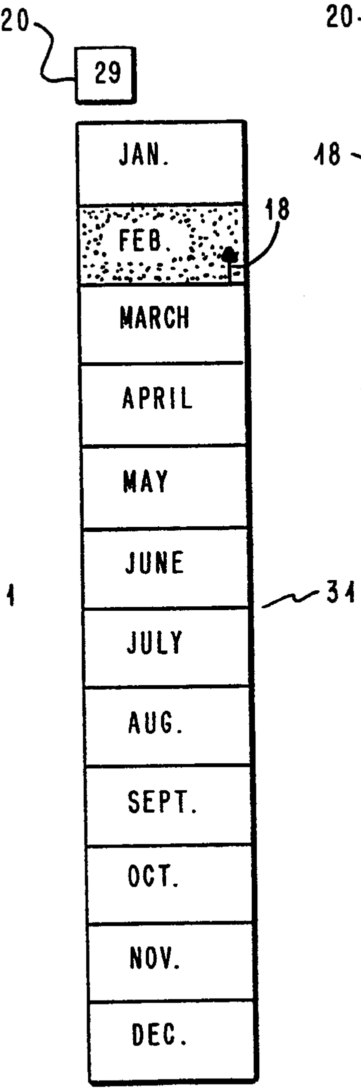


FIG. 7B

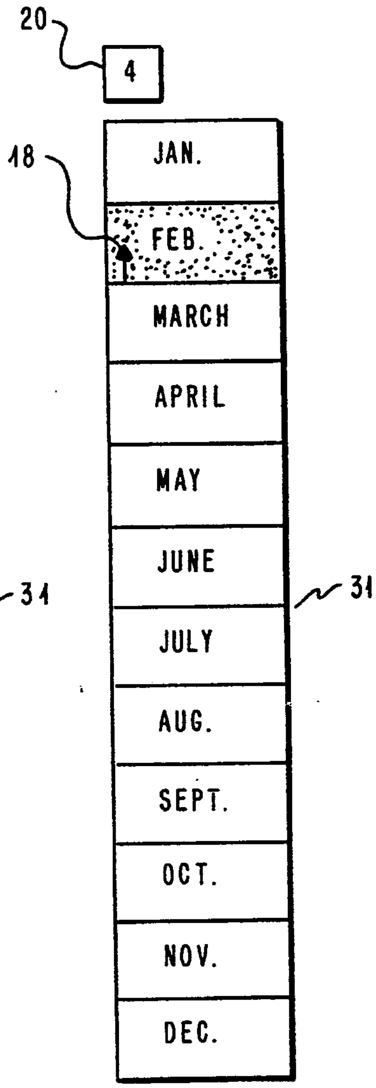


FIG. 7C